Instance Selection in Text Classification Using the Silhouette Coefficient Measure

Hugo Jair Escalante

Advances in Artificial Intelligence

Instance Selection in Text Classification Using the Silhouette Coefficient Measure

Debangana Dey [1], Thamar Solorio [1], Manuel Montes y Gomez [1,2],
and Hugo Jair Escalante [2,3]

[1] Department of Computer and Information Sciences
University of Alabama at Birmingham
1300 University Boulevard, Birmingham AL 35294.
[2] National Institute of Astrophysics, Optics and Electronics
[3] Luis Enrique Erro No. 1, Sta. Maria Tonantzintla, Puebla, CP: 72840, Mexico.
[4] Universidad Autonoma de Nuevo Leon
San Nicolas de los Garza, 66450, N. L. Mexico.
{deb19,solorio}@uab.edu,
{mmontesg,hugojair}@ccc.inaoep.mx

Abstract. The paper proposes the use of the Silhouette Coefficient (SC) as a ranking measure to perform instance selection in text classification. Our selection criterion was to keep instances with mid-range SC values while removing the instances with high and low SC values. We evaluated our hypothesis across three well-known datasets and various machine learning algorithms. The results show that our method helps to achieve the best trade-off between classification accuracy and training time.

Keywords: Instance Selection, Outlier Elimination, Text Classification, Supervised Machine Learning.

1 Introduction

Text classification is the problem of assigning the most appropriate label to a new document. For the task of automated text classification based on machine learning algorithms, we need to develop a manually annotated training set in order to train the classifiers. During this developmental phase, three major problems are encountered. First, the training sets are highly vulnerable to human errors of judgment leading to mislabeling of the instances in the training corpus. Second, data ambiguity is also a reason that contributes to the confusion in the labeling of documents. For example, if a news article reports, "The President was present at the inaugural ceremony of the soccer game" it becomes difficult to categorize it as a "sports" article or as a "political" article. Such misclassified instances are noisy. Third, there are certain instances that contain redundant content and the inclusion of each one of them in the training set costs more computational time without bringing any new knowledge to the classifier. The presence of such instances causes redundancy in the training set. In automated text classification, a classifier's accuracy is affected by the presence of the

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part I, LNAI 7094, pp. 357–369, 2011.

noisy data in the training corpus. Therefore, our aim is to improve the performance of a classifier in order to achieve the most acceptable rate of accuracy by eliminating the noisy and redundant instances from the training set.

In text categorization tasks, the most common way to represent data is by building a vector space model of documents and their features. The document collection is transformed into a matrix where the documents make the rows and the features make the columns. The weights represent the relevance of the features in the documents. The set of features is generally the vocabulary of the training dataset. But when dealing with a large dataset, the processing of tens and thousands of instances and features often leads to increased computational times, difficulty in maintenance and manipulation of the data structures, hamper comprehensibility of the learned model and often leads to infeasibility in performing the task. Thus, an important step is to reduce the dimensionality of the data structures, by doing instance selection or feature selection in order to preserve the relevant information and reduce computational costs.

Through *Instance Selection*, an attempt is made to get rid of non-useful instances such that acceptable or higher accuracy is obtained at considerably reduced runtimes. The goal of instance selection can be focused on two main approaches: removal of instances that are undesirable, and selection of instances that are more informative. The first approach can be assumed to encompass the concept of *outlier elimination,* where an outlier is an object in a class that is so different from the other objects in the same class that it seems to have been generated by some other mechanism. The second approach focuses on the selection of instances that are not redundant and can contribute unique knowledge to the learning model. In instance selection, the primary intention is to keep good examples of a class that either help to represent the class firmly or distinguish it from the other classes. The concept of instance selection has taken a new shape in the form of 'Active Learning' in some of the recent works [1]. Active learning is a way to select useful instances to be labeled by an oracle or a human annotator from hundreds and even thousands of unlabeled instances. The goal of active learning is to reduce the cost of manually annotating the data by selecting the most informative instances. This approach is useful for machine learning problems where obtaining labeled data is very expensive.

In some of the previous works, the focus was mainly on the selection of instances in an object classification scenario [2,3,7,13]. The challenges faced in a real world text categorization problem are the numerous class labels and the huge amount of training data required to train a classifier. This in turn leads to massive data structures that demand powerful computational resources. As such, the need to find a way to reduce the size of these data structures so that they are manageable and a way to require of less number of labeled instances that are expensive to obtain, becomes inevitable. Thus, in this work, we have tried to attain the goal of Instance Selection in a text classification setting making use of the *Silhouette coefficient* measure, which is an evaluation technique for clustering algorithms [2,19]. We propose to compute the silhouette coefficient for each instance in the training set and then to eliminate the instances with high and low values. In other words, we propose eliminating the border and core instances in order to retain the instances that show a good trade-off between descriptive and discriminative power. In order to show the performance of our proposed method, we experimented with three popular text collections in the classification domain: Reuters R8, 20 Newsgroup and the Classic4 collection. Our results show that with decreasing number of training instances, our method of retaining mid-range instances achieves a higher accuracy at an acceptable period of training time, than the accuracy obtained when high or low SC valued instances were retained.

The paper is organized as follows: in Section 2, we present some related work followed by the details of our proposed method in Section 3. We have stated some details about our datasets in Section 4. The experimental settings are outlined in Section 5 and we present our results in Section 6. We have concluded our work in the last section.

2 Related Work

In his work, Czarnowski uses the similarity coefficient to identify clusters of instances in one of his object selection approaches and uses the Silhouette Coefficient measure to evaluate the cluster quality of each of his methods [3]. A single instance was selected as a prototype of the class and used for training purposes. He uses agent-based population learning algorithms to select the desired prototypes from good quality clusters. He also uses clustering approaches for feature selection, known as *one-way clustering*. Using this method, he replaces the original feature space by clusters of features. The clusters are made based on similarity between the features. Some of the important one-way clustering methods applied in previous work are: information bottleneck [4], distributional clustering [5], and divisive clustering [6].

Lopez et al. use clustering of training instances to determine border objects in each class to build a set of prototypes to be used for classification [7]. Along with the border instances they have also included in their OSC (Object Selection by Clustering) method the centroids for each class as a core representative of the class. The intuition behind keeping border instances is to maintain the discriminative characteristics of each class, which in turn reduces the importance of core instances as they become superfluous to the class, storing redundant information [8,9]. Through their experimental evaluation of the various object selection methods like CLU method (CLUster) [10] and the DROP (Decremental Reduction Optimization Procedure) [8], they concluded that the CLU and DROP methods performed fairly well with the classifiers based on nearest neighbor rule, whereas their proposed method, OSC, performed well with the classifiers other than nearest neighbor classifiers. This was probably due to the fact that DROP and CLU are based on the nearest neighbor rule.

Another concept is that of outlier elimination, whereby the border objects are considered to be those instances that contain confusing content harmful for the classification task. The reference frame, relative to which the outliers are to be determined, can be either global or local. A local method considers only the information embedded in a document's own class, whereas a global method utilizes information extracted from the other classes of the corpus and not only the document's own class. Shin et al. in their work approached the outlier elimination problem by using a local method [11]. They first calculated a centroid for each of the classes in the training corpus, and then imposed a threshold radius around each of those centroids. By calculating distances using the cosine similarity measure, whichever document lied at a distance greater than or equal to that threshold radius, was considered to be an outlier and was eliminated. This process was conducted for each of the classes in the corpus. Finally, this refined corpus was used to train a centroid based classifier. Their results showed that their classifier works better in the latter scenario.

In certain approaches, clusters of similar instances are formed and instances close to the clusters are considered as prototypes [12]. The algorithms chosen to compute the centroids can affect their quality and thus, have an impact on the prototype selection. Again, for large datasets, when the number of features is large, ''the quality of the centroids can be poor" [3].

Lopez et al. in their work have adopted a simple, local method for prototype selection [13]. They have proposed a PSR (Prototype Selection by Relevance) method that computes the relevance weight for each instance or prototype within a class by averaging the similarity of that instance with all the other instances in its own class. A certain number of instances with the highest relevance weights have been retained. Among the retained instances, they have considered a few to be border elements. They have compared their method with other methods like DROP3, DROP5 (Different variations of DROP) and GCNN (Generalized Condensed Nearest Neighbor Rule). Their method could not outperform the other methods in case of smaller datasets but the advantage of runtime gave PSR an edge over the other methods when dealing with large datasets.

Previous works on instance selection have focused on local or global methods for the general problem of object classification. Text Classification is a relatively more challenging area of research because of the large number of classes and training sets. In our work, we propose the use of the *Silhouette Coefficient (SC)* [2] measure to rank instances for selection in the problem of Text Classification. Like Lopez et al.'s work [14] we have also tried to avoid the use of clustering techniques because clustering consumes more time to process large datasets. Moreover, our method is a global method and does not follow the nearest neighbor rule. Unlike Lopez et al.'s work, we have discarded the border and core instances to retain instances with medium relevance as depicted by their SC values. To the best of our knowledge, we are the first to evaluate this criterion of instance selection in a text classification scenario. Our intuition was that these instances represent a good trade-off between descriptive and discriminative power and, therefore, that they are the most informative for classification purposes. In addition, we have applied our method on large datasets of text documents that pose a great challenge to maintain the huge number of features generated. Our method shows enhanced accuracy and faster runtime over the dataset, reduced by our method. Taking accuracy and runtime combined as a tuple for performance evaluation we found our method to be promising.

3 Proposed Method

Our proposed method consists of ranking the instances to select those that are deemed to be more helpful in the classification task. We assume that in a corpus where instances are ranked based on a global relevance value, instances with higher relevance will be located at the *core* of any class, and those with lower relevance will lie at the *peripheral* locations of a class, since they are closer to the other classes. In such a scenario, we propose to eliminate the instances with high or low relevance values to retain what we call the *mid-ranked instances*. The motivation relies on a concept similar to term selection by transition point where mid-frequency terms are assumed to have high semantic information that can be useful in indexing the documents [25]. Based on Zipf's Law of Word Occurrences [22], refined concepts of Booth [23], and Urbizagástegui [24], transition point is a threshold frequency value computed in such a way that it can split the document vocabulary in a high or low frequency set. Pinto et. al. have used this technique to cluster documents of a corpus with narrow domain and short texts [20].

To evaluate the relevance of the training instances we propose the use of Silhouette Coefficient. In clustering tasks the *SC* width is calculated for each of the instances in the clusters in order to evaluate the cluster solution. Through this evaluation technique the following can be computed: SC value for each instance of the dataset, an average SC value for each cluster of the dataset, and an overall SC value for the whole dataset. The SC values are very helpful in denoting the cohesiveness of the instances in one cluster and the separation of instances in one cluster from those in the other clusters. Moreover, the SC measures are helpful in judging the effectiveness of a clustering algorithm.

Consider a document collection *D* having *n* classes, and let $|c_k|$ denote the number of documents from the k^{th} class of the corpus, and $dist(d_i, d_j) = 1 - \cos(d_i, d_j)$ indicate the distance between documents d_i and d_j. The *SC* value for the document d_i is computed by using the following formula:

$$SC_D(d_i \in c_k) = \frac{\big(b(d_i) - a(d_i)\big)}{max\big(b(d_i), a(d_i)\big)} \tag{1}$$

where:

$$a(d_i \in c_k) = \frac{1}{|c_k| - 1} \sum_{\substack{\forall d_j \in c_k \\ d_j \neq d_i}} dist\big(d_i, d_j\big) \tag{2}$$

$$b(d_i \in c_k) = \min_{j \neq k} \left[\frac{1}{|c_j|} \sum_{\forall d_m \in c_j} dist\,(d_i, d_m) \right] \tag{3}$$

In Equation 2, we calculate the average distance of the document d_i with all the documents in its own class. This average distance becomes the *a-value* of d_i. In Equation 3, we calculate the average distance of d_i with the documents of the other classes in the corpus. We then consider the minimum of all these average values to denote the *b-value* for d_i. The SC values range from -1 to +1. Therefore, if a document has SC value near to +1, it means that for d_i , *a-value* < *b-value*.

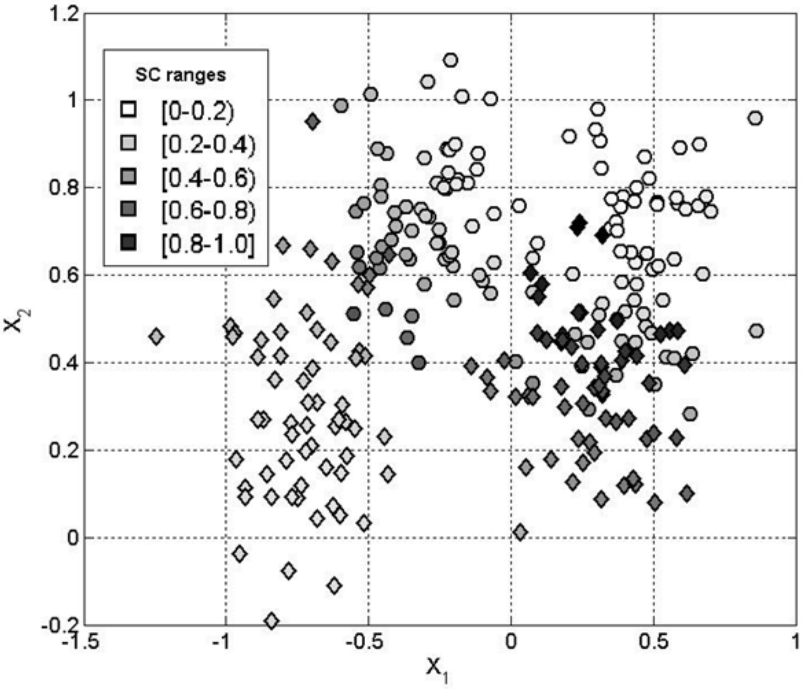


Fig. 1. Shows the SC values for a two-class syntectic data set with two attributes (X_1 and X_2), the number of classes and dimensionality were chosen for clarity

Figure 1 shows the SC value ranges for the data points in a syntectic dataset with a two-dimensional attribute space. Through the figure, we can clearly see that instances with low SC values (light gray) are those difficult to classify and can be considered noisy or potential outliers. In text categorization these instances correspond to texts that may contain information from different thematic categories and hence they can be removed. On the other hand, instances with high SC values (dark gray) are examples that can be considered easy to classify. In text categorization these instances correspond to prototypical texts of each category, although we can see that they can also be redundant and, therefore, eliminating a portion of these instances is expected to not hurt classification accuracy.

4 Datasets

The datasets used for our experiments are the following:

- Reuters-21578 R8 Collection- This dataset has 8 classes, 5,485 documents in the training set and 2,189 documents in the test set. The documents are non-uniformly distributed over the classes [17].
- Classic4 Collection- This dataset has 4 classes and 7,095 documents. We split the dataset randomly to obtain 60% documents as training set and 40% as test set [16].
- 20 Newsgroup Collection- This dataset consists of 20 classes and 18,828 documents. This dataset again was randomly split by us into 60% training and 40% testing set [18].

Each of the datasets was preprocessed for our classification task. We eliminated words from the feature set that appeared in a standard stop-list.

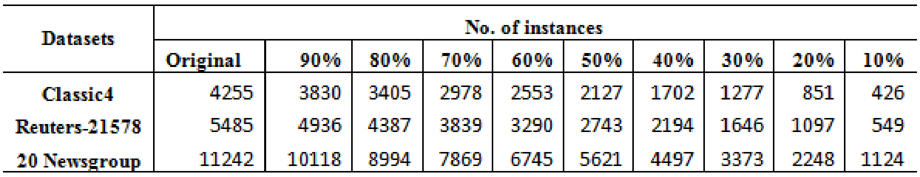
5 Experimental Settings

We have represented our documents using the standard vector space model where the features have formed the columns and the documents have formed the rows. The *term frequency-inverse document frequency* (tf-idf) weights have been used to fill up the term-document matrices.

For each experiment we have retained different number of text documents to determine the minimal number of training instances required for that dataset to achieve acceptable accuracies. By doing this we are also trying to dynamically determine a threshold for each dataset, such that the peripheral and the core instances can be identified. In all our experiments we start with the original training set and remove instances incrementally by steps of 10%. Thus, we end up with training sets of 90%, 80%, 70% and so on until we only have 10% on the training set. Our evaluation metric is accuracy computed as a percentage of test instances correctly classified by the algorithms.

In Table 1 we show the number of instances in each configuration for each of our data sets. The test data sets remain unchanged for all configurations; see Section 4 for a description of these sets.

Table 1. Number of instances retained for each experiment



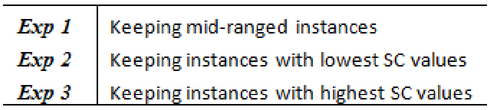
Datasets	No. of instances									
	Original	90%	80%	70%	60%	50%	40%	30%	20%	10%
Classic4	4255	3830	3405	2978	2553	2127	1702	1277	851	426
Reuters-21578	5485	4936	4387	3839	3290	2743	2194	1646	1097	549
20 Newsgroup	11242	10118	8994	7869	6745	5621	4497	3373	2248	1124

We have tested our method using support vector machines (SVM), nearest neighbors (kNN), Naïve Bayes (NB) and random forest (RF) as the base classifiers. We have also used the Weka toolkit that implements the above classifiers [15]. As mentioned previously, our goal is to evaluate our idea of keeping instances with mid-range relevance values in terms of classification accuracy and running time. Thus, we report the difference in classification accuracy with respect to using the entire training set and running time for all three data sets.

6 Results

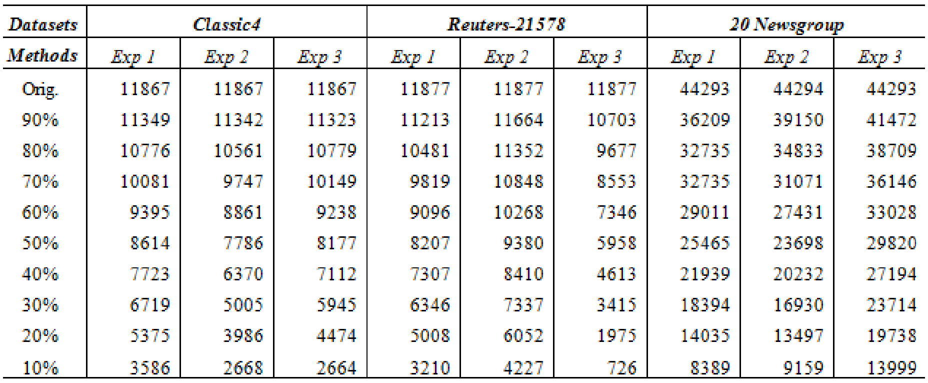
Table 2 shows the different experiments we performed. The first experiment (Exp 1) represents our idea of removing outliers and core instances by incrementally removing the same number of instances with the lowest and highest relevance values. For example, in a training set with only 80% of the instances, we remove 10% of the instances with the highest relevance values and 10% of the instances with the lowest values. However, for comparison purposes we also performed experiments with two different selection criteria: one where we keep instances with the lowest SC values (Exp 2), that is the border objects as in previous work, and one where we keep instances with the highest SC values (Exp 3), the core instances, again as previous work has done.

Table 2. List of the experiments performed



Exp 1	Keeping mid-ranged instances
Exp 2	Keeping instances with lowest SC values
Exp 3	Keeping instances with highest SC values

Table 3. The number of attributes extracted from the training set during each experiment



Datasets	*Classic4*			*Reuters-21578*			*20 Newsgroup*		
Methods	*Exp 1*	*Exp 2*	*Exp 3*	*Exp 1*	*Exp 2*	*Exp 3*	*Exp 1*	*Exp 2*	*Exp 3*
Orig.	11867	11867	11867	11877	11877	11877	44293	44294	44293
90%	11349	11342	11323	11213	11664	10703	36209	39150	41472
80%	10776	10561	10779	10481	11352	9677	32735	34833	38709
70%	10081	9747	10149	9819	10848	8553	32735	31071	36146
60%	9395	8861	9238	9096	10268	7346	29011	27431	33028
50%	8614	7786	8177	8207	9380	5958	25465	23698	29820
40%	7723	6370	7112	7307	8410	4613	21939	20232	27194
30%	6719	5005	5945	6346	7337	3415	18394	16930	23714
20%	5375	3986	4474	5008	6052	1975	14035	13497	19738
10%	3586	2668	2664	3210	4227	726	8389	9159	13999

Tables 4 to 7 show the results for our three collections. We show the accuracy (in percentage) reached by different learning algorithms with the original training set, and the change in accuracy after instance reduction. The tables also report the change in accuracy averaged across all the learning algorithms.

Table 4. The 'Change in Accuracy' observed when Exp 1 (see Table 2) was conducted

Datasets	*Classic4*					*Reuters-21578*					*20 Newsgroup*				
Algorithms	SMO	KNN	NB	RF	**Avg.**	SMO	KNN	NB	RF	**Avg.**	SMO	KNN	NB	RF	**Avg.**
Orig.	91.26	47.04	70.79	85.80	**73.72**	94.61	62.77	86.39	87.76	**82.88**	82.13	28.97	60.00	44.54	**53.91**
90%	0.25	-0.78	13.78	0.04	**3.32**	-0.55	-1.74	0.13	-0.14	**-0.57**	-0.67	-4.31	1.57	-1.69	**-1.28**
80%	0.11	-1.16	15.22	1.20	**3.84**	0.41	-3.15	0.64	0.18	**-0.48**	-3.01	-7.90	0.70	-4.37	**-3.65**
70%	-0.32	-1.13	22.41	1.76	**5.68**	0.18	-4.52	0.54	0.82	**-0.74**	-4.90	-11.44	0.39	-8.32	**-6.07**
60%	-0.56	-1.16	21.88	0.32	**5.12**	0.27	-4.98	1.05	0.27	**-0.85**	-7.09	-13.76	-1.65	-8.86	**-7.84**
50%	-0.81	-1.23	21.21	1.69	**5.21**	0.14	-6.44	1.60	-0.64	**-1.34**	-9.96	-16.12	-3.52	-7.93	**-9.38**
40%	-2.78	-1.41	20.19	-0.11	**3.97**	-0.82	-7.58	1.60	-1.64	**-2.11**	-14.35	-18.10	-5.21	-15.36	**-13.25**
30%	-5.32	-1.55	18.36	-4.72	**1.69**	-1.92	-8.41	0.77	-4.39	**-3.49**	-19.74	-18.98	-8.40	-19.99	**-16.78**
20%	-6.38	-1.73	16.63	-6.55	**0.49**	-6.30	-9.59	-0.37	-7.77	**-6.01**	-26.79	-20.94	-12.56	-21.76	**-20.51**
10%	-10.32	-1.76	10.64	-10.32	**-2.94**	-19.32	-11.42	-3.48	-13.70	**-11.98**	-40.56	-22.55	-21.25	-30.05	**-28.60**

Table 5. The 'Training Time (seconds)' required when our method (Exp 1, see Table 2) was applied

Datasets	***Classic4***			***Reuters-21578***			***20 Newsgroup***		
Algorithms	SMO	NB	RF	SMO	NB	RF	SMO	NB	RF
Orig.	44.3	72.69	110.58	52.78	87.16	119.97	217.34	630.29	674.36
90%	31.99	58.97	95.08	38.13	73.52	107.92	135.13	503.8	513.48
80%	22.91	47.12	74.39	28.63	59.64	81.59	119.23	427.84	449.31
70%	18.25	42.36	65.83	21.49	48.86	66.89	96.76	306.68	305.91
60%	13.19	33.14	52	17.25	38.23	52.95	77.7	211.38	254.52
50%	9.06	23.98	41.2	11.2	28.21	38.61	46.26	154.16	161
40%	6.5	17.17	27.37	10.31	19.66	26.67	29.4	97.35	103.69
30%	3.95	11.38	16.92	5.27	12.31	16.61	14.44	59.38	61.71
20%	1.73	5.52	9.03	2.78	6.27	9.47	7.23	21.49	29.81
10%	0.5	1.72	2.95	1.02	1.88	2.84	2.59	7.88	10.06

Table 5 reports the training times recorded for each combination of machine learning algorithms and the subsets of data. The training time recorded for kNN algorithm was omitted since this algorithm does not include any training phase.

Table 6. The 'Change in Accuracy' observed when Exp 2 (see Table 2) was conducted

Datasets	*Classic4*					*Reuters-21578*					*20 Newsgroup*				
Algorithms	SMO	KNN	NB	RF	**Avg.**	SMO	KNN	NB	RF	**Avg.**	SMO	KNN	NB	RF	**Avg.**
Orig.	91.26	47.04	70.79	85.80	**73.72**	94.61	62.77	86.39	87.76	**82.88**	82.13	28.97	60.00	44.54	**53.91**
90%	0.00	-1.06	0.32	-2.54	**-0.82**	0.05	-1.46	-0.28	-0.96	**-0.66**	-0.55	-5.05	0.87	-2.60	**-1.83**
80%	-0.28	-1.23	-0.63	-3.10	**-1.31**	0.09	-2.33	-0.78	0.82	**-0.55**	-2.37	-8.67	-0.68	-6.61	**-4.58**
70%	-1.73	-1.52	-3.56	-12.33	**-4.78**	-0.09	-2.97	-0.87	-0.96	**-1.22**	-4.88	-11.76	-4.56	-13.44	**-8.66**
60%	-4.40	-1.73	-4.55	-18.71	**-7.35**	-0.91	-4.39	-1.15	-3.43	**-2.47**	-11.68	-14.08	-9.46	-18.75	**-13.49**
50%	-13.46	-1.80	-7.33	-25.97	**-12.14**	-7.35	-1.83	-10.60	-9.91	**-7.42**	-23.03	-16.16	-16.41	-23.26	**-19.72**
40%	-30.58	-1.90	-22.34	-35.76	**-22.65**	-15.44	-1.23	-24.49	-24.12	**-16.32**	-35.11	-19.02	-22.99	-26.80	**-25.98**
30%	-51.37	-1.90	-37.84	-40.03	**-32.79**	-29.56	-14.02	-20.83	-30.61	**-23.75**	-48.08	-20.27	-29.80	-30.34	**-32.12**
20%	-55.25	-1.90	-44.54	-40.42	**-35.53**	-33.90	-15.12	-23.48	-30.61	**-25.78**	-59.48	-21.56	-36.81	-32.15	**-37.50**
10%	-50.35	-1.87	-50.53	-40.59	**-35.84**	-29.97	-12.33	-33.90	-22.48	**-24.67**	-66.77	-23.04	-42.89	-35.20	**-41.98**

Table 7. The ‘Change in Accuracy’ observed when Exp 3 (see Table 2) was conducted

Datasets	*Classic4*					*Reuters-21578*					*20 Newsgroup*				
Algorithms	SMO	KNN	NB	RF	**Avg.**	SMO	KNN	NB	RF	**Avg.**	SMO	KNN	NB	RF	**Avg.**
Orig.	91.26	47.04	70.79	85.80	**73.72**	94.61	62.77	86.39	87.76	**82.88**	82.13	28.97	60.00	44.54	**53.91**
90%	-0.32	-0.18	14.66	0.60	**3.69**	1.14	-3.06	0.45	0.87	**-0.15**	-2.25	-3.69	-0.43	-1.18	**-1.89**
80%	-0.63	-0.21	21.74	2.22	**5.78**	0.05	-4.39	0.36	0.05	**-0.98**	-4.44	-7.00	-3.77	-1.97	**-4.29**
70%	-3.38	-0.46	19.56	-12.47	**0.81**	-1.28	-6.35	0.59	-1.74	**-2.19**	-6.58	-21.11	-6.75	-4.89	**-9.83**
60%	-7.47	-0.81	17.58	-2.96	**1.59**	-6.26	-7.50	0.50	-9.91	**-5.79**	-9.41	-11.67	-7.30	-5.56	**-8.48**
50%	-8.28	-0.95	16.95	-4.44	**0.82**	-31.48	-9.41	-20.06	-27.91	**-22.21**	-11.87	-13.94	-7.85	-6.79	**-10.12**
40%	-10.50	-1.23	15.50	-6.34	**-0.64**	-33.44	-9.87	-24.22	-28.41	**-23.99**	-19.88	-13.52	-15.06	-9.55	**-14.50**
30%	-14.06	-1.44	14.13	-10.15	**-2.88**	-33.49	-10.46	-25.27	-29.65	**-24.72**	-26.68	-16.56	-15.64	-13.37	**-18.06**
20%	-21.35	-1.62	10.68	-16.21	**-7.13**	-39.06	-12.11	-25.87	-34.35	**-27.85**	-37.32	-19.28	-18.09	-19.30	**-23.49**
10%	-35.91	-1.69	-0.28	-24.70	**-15.64**	-44.45	-13.11	-30.61	-37.87	**-31.51**	-53.72	-17.13	-26.50	-25.12	**-30.62**

The results on these tables show a consistent trend, removing instances with highest and lowest relevance values yields the best trade-off between accuracy and run time. For instance, see the results for keeping only 50% of the training instances. For all three collections the smallest loss in accuracy is reached by our method, and for some learning algorithms we even gain accuracy, see results for NB in Table 4. This trend also hints at the fact that instance-based learning algorithms benefit the most from instance reduction.

Another consistent and relevant trend in our results is that if we do an extreme instance reduction, by keeping only 10% of the training instances, the loss in accuracy overall will be less when using our approach than any of the other two. By comparing the results reported in Table 4 and 7, it is clear that for a limited number of instances, such as just 10% of the total corpus, selection of mid-ranged instances for training is a better idea than instances with either high and low relevance. This is encouraging and a very practical finding since if computational resources are very expensive, and lower accuracy rates are acceptable, our approach will yield the best trade-off between efficiency and effectiveness. Our results show that at times it is also possible to achieve higher accuracy than the baseline, when the mid-ranged instances are retained, whereas, if we remove these instances, the accuracy values continue to decrease consistently with a shrinking training set, for all the learning algorithms. Therefore, it is evident that the mid-ranged instances carry important information concerning the class description and cannot be ignored.

These results also show some effects on the differences among the corpora. For instance, in the Classic4 collection, if we remove only 20% of the training instances, we obtain slightly better results by removing the ones with lowest relevance values. This leads us to believe that Classic4 has more outliers in the training set than the other two collections. The other two collections still show better results when using our method.

Lastly, it is very interesting to see that a secondary effect of our instance reduction approach is feature reduction. All tables show in the last row the resulting number of features. As can be seen in the last row of the tables, we reach a considerable reduction in the number of features. This is an effect particular to instance selection in text classification, because after removing instances, a lot of the words present in those documents are not present in the other documents and can be removed from the feature vectors. We have not seen a related discussion about this in any previous work. It would be interesting to explore further how this method compares to other approaches on feature selection. This however, is beyond the scope of this work and we will leave it for future work.

7 Conclusions

In this paper we have proposed the use of the Silhouette Coefficient measure for the purpose of determining the most effective set of training instances in text classification. This measure is typically used to evaluate the quality of clusters generated after the application of clustering algorithms on the data. We have calculated the SC measure for each of the instances in the training set. We then rank the instances according to their SC measures; those with the lowest and highest SC values were eliminated, assuming them to be noisy or to contain redundant information that might not aid us in our classification task. We found that instances in the middle range of SC values were capable of retaining more descriptive information about their respective classes than instances with extreme SC values.

We have taken accuracy values and training time as a tuple for performance evaluation. Elimination of the instances in the training set led to a drop in accuracy from the original training set in most of the cases. But the consistent tradeoff between accuracy measures and training time was commendable when our method was applied. There were also cases when we observed a higher accuracy when instances with medium ranks were retained in the training sets. Unlike the peripheral or core instances, the medium ranked instances proved to provide more information that could help in describing their classes and also discriminate them from each other. We have evaluated our results by carrying out experiments where only peripheral instances were eliminated or only core instances were removed. Training with only the peripheral instances or only the core instances led to a higher drop in the accuracy values than when medium-ranked instances were used to train the model. The experimental results for all the three datasets showed the same trend of maintaining an acceptable accuracy coupled with considerable reduction of time.

The Silhouette Coefficient measure seemed to be an interesting parameter for evaluation and we would like to delve deeper into its analysis and try to find out what other information can be extracted with the use of this measure. We would like to test and compare our method using different datasets having narrow domain unlike the broad domain datasets used in this work. Also, it will be interesting to investigate the efficacy of our method in the scenario of short text classification using the abridged versions of the datasets used for this work. Lastly, we would like to propose other measures to determine relevant instances for training classifiers in a text classification setting.

Acknowledgments. This work was supported in part by the CONACYT-Mexico (project no. 134186) and by the European Commission as part of the WIQ-EI project (project no. 269180) within the FP7 People Programme.

References

1. Settles, B.: Active Learning Literature Surve, Computer Sciences Technical Report 1648. University of Wisconsin–Madison (2009)
2. Hubert, M., Struyf, A., Rousseeuw, P.: Clustering in an Object-Oriented Environment. Journal of Statistical Software 1(4) (1996)
3. Czarnowski, I.: Cluster-based instance selection for machine classification. Knowledge and Information Systems (2011)
4. Slonim, N., Tishby, N.: Agglomerative Information Bottleneck. In: Advances in Neural Information Processing systems (NIPS-12) (1999)
5. Baker, L.D., McCallum, A.K.: Distributional clustering of words for text classification. In: Proc. of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR 1998, pp. 96–103 (1998)
6. Dhillon, I.S., Mallela, S., Kumar, R.: A Divisive Information-Theoretic Feature Clustering Algorithm for Text Classification. Journal of Machine Learning Research 3, 1265–1287 (2003)
7. Olvera-López, J.A., Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F.: Object selection based on clustering and border objects. In: Kurzynski, M., et al. (eds.) Computer Recognition Systems 2, Wroclaw, Poland. ASC, vol. 45, pp. 27–34 (2007)
8. Martinez, T.R.: Reduction Techniques for Instance-Based Learning Algorithms. Machine Learning 286 (2000)
9. Brighton, H.: Advances in Instance Selection for Instance-Based Learning. Algorithms. Knowledge Creation Diffusion Utilization, 153–172 (2002)
10. Lumini, A., Nanni, L.: A clustering method for automatic biometric template selection. Pattern Recognition 39, 495–497 (2006)
11. Shin, K., Abraham, A., Han, S.-Y.: Enhanced Centroid-Based Classification Technique by Filtering Outliers. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2006. LNCS (LNAI), vol. 4188, pp. 159–163. Springer, Heidelberg (2006)
12. Wilson, D.R., Martinez, T.R.: An Integrated Instance-Based Learning Algorithm. Computational Intelligence 16, 1–28 (2000)
13. Olvera-López, J.A., Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F.: Prototype selection via prototype relevance. In: Ruiz-Shulcloper, J., Kropatsch, W.G. (eds.) CIARP 2008. LNCS, vol. 5197, pp. 153–160. Springer, Heidelberg (2008)
14. Kira, K., Rendell, L.A.: A practical approach to feature selection. In: Proc. of 9th International Conference on Machine Learning, pp. 249–256 (1992)
15. Hall, M., Frank, E., Holmes, G., Bernhard, P., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations 11(1) (2009)
16. Classic4 Dataset, `ftp://ftp.cs.cornell.edu/pub/smart/`
17. Reuters R8 Dataset, `http://www.daviddlewis.com/resources/testcollections/reuters21578`
18. 20 Newsgroup Dataset, `http://people.csail.mit.edu/jrennie/public_html/20Newsgroup/`
19. Rousseeuw, P.J.: Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis. Computational and Applied Mathematics 20, 53–65 (1987), doi:10.1016/0377-0427(87)90125-7
20. Pinto, D., Rosso, P., Jiménez-Salazar, H.: A Self-enriching Methodology for Clustering Narrow Domain Short Texts. Computer Journal 54(7), 1148–1165 (2011)

21. Pinto, D., Rosso, P., Jiménez-Salazar, H.: A Self-enriching Methodology for Clustering Narrow Domain Short Texts. Computer Journal 54(7), 1148–1165 (2011)
22. Zipf, G.K.: Human Behavior and the Principle of Last-Effort. Addison-Wesley, Cambridge (1949)
23. Booth, A.: A Law of Ocurrences for Words of Low Frequency. Information and Control (1967)
24. Urbizagástegui, R.: Las posibilidades de la Ley de Zipf en la indización automática, Research report of the California Riverside University (1999)
25. Pinto, D., Jiménez-Salazar, H., Rosso, P.: Clustering Abstracts of Scientific Texts Using the Transition Point Technique. In: Gelbukh, A.F. (ed.) CICLing 2006. LNCS, vol. 3878, pp. 536–546. Springer, Heidelberg (2006)